

TP N° 7

Intégration continue d'une API Java avec Jenkins

Il s'agit de développer le processus d'intégration continue d'une API. La figure ci-dessous représente le pipeline.



1 Création du projet

1. Créez un projet Gradle et intégrez le dossier *src*, le fichier *build.gradle*, le dossier *feature* et le fichier *Jenkinsfile* dans la racine du projet (garder dans le fichier *build.gradle* les plugins de Jacoco, Sonar et maven-publish).
2. Partagez le code source sur le GitHub.
3. Après le lancement de Jenkins et l'installation des différents plugins, créez un *Multibranch Pipeline* project (voir le document de configuration).
4. Ajoutez le webhook de l'instance Jenkins dans le repository GitHub (voir le document de configuration).

2 Les phases du pipeline

2.1 La phase Test

Cette phase est composée des étapes suivantes:

1. Lancement des tests unitaires.
2. Archivage des résultats des tests unitaires.
3. Génération des rapports de tests Cucumber.

2.2 La phase Code Analysis

Cette phase consiste à analyser la qualité du code avec SonarQube.

2.3 La phase Code Quality

Dans cette phase, l'état de Quality Gates doit être vérifié. Si cette dernière est dans l'état *Failed*, l'exécution du pipeline doit s'arrêter.

2.4 La phase Build

Cette phase est composée des étapes suivantes:

1. Génération du fichier Jar.
2. Génération de la documentation.
3. Archivage du fichier Jar et de la documentation.

2.5 La phase Deploy

Cette phase consiste à déployer le fichier Jar généré dans <https://mymavenrepo.com/>.

2.6 La phase Notification

Les notifications concernent l'envoi d'un message informant l'équipe de développement du déploiement avec succès: par mail et sur *Slack*. En cas d'échec dans l'une des phases du pipeline, une notification doit être envoyée pour informer l'équipe de développement.